Volume 4, Issue 8, August 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Making Cloud Computing More Efficient

Amanpreet Kaur
Amity School of Engg. & Tech.
Noida, India

Deepa Gupta
Amity Institute of I.Tech.
Noida, India

Deepak Kumar Verma
IEC College of Engg. & Tech.
G. Noida, India

Abstract— Green computing is a study and practice of designing, manufacturing, using of Computers, servers & associated systems (such as storage devices, monitors, printers and networking & communications systems) efficiently & effectively with minimal impact on the environment. Study continues into key areas such as making computers as energy efficient as possible & designing algorithms & hardware for efficiency related computer technologies. A few key approaches being discussed here,

- ***Scalability - the best solution to increasing and maintaining application.***
- ***Green cloud computing Architecture.***
- ***Virtual machine migration***
- ***Power management with Dynamic Voltage & Frequency Scaling (DVFS)***

Keywords— Green Cloud computing, Performance of Cloud, Scalability of cloud, Cloud Throughput, Resource management for cloud.

I. INTRODUCTION

Cloud computing [1-3] is blend of old-fashioned computing equipment and network technology like distributed /grid /parallel computing. Cloud Computing is a universal term used to describe a new class of network based computing that takes place over the Internet. Essentially a group of integrated & networked hardware, software and Internet infrastructure for communication and networking services to clients.

Cloud computing [3] is one of the hottest topic as an emerging new computing model. It is style of computing in which dynamically scalable & other virtualized assets are provided as a package over the Internet. It is the old-fashioned network technology comprising parallel computing, distributed computing, utility computing, virtualization, network storage technologies, load balance combined with other products. Cloud computing is a model for simplifying universal, on-demand network access to a common pool of configurable computing assets by setting up basic hardware and software arrangements in a data centre.

The aim of green cloud computing [7] is to plan a high performance, low-power computing infrastructure while adapting an energy-efficient and safe service mode. As companies shift computing resources from premises-based data centres to private and public cloud computing services, they should make certain their applications and data make a safe and smooth evolution to the cloud. In particular, companies should ensure cloud-based facilities will deliver necessary application and transaction performance—now, and in the future. Much depends on this immigration and homework for the transition and ultimate cut-off. Rather than merely moving applications from the old-style data centre servers to a cloud computing environment and click the “on” switch, enterprises should scrutinize performance issues[3], prospective reprogramming of applications and capacity planning to completely heighten application performance. Applications that performed one way in the data centre may not perform identically on a cloud platform.

Enterprise need to insulate the areas of an application or its deployment that may cause performance changes and address each separately to assure optimal transition. In many cases, however, the fundamental infrastructure of the cloud platform may directly affect application performance. Businesses should also meticulously test applications developed and deployed explicitly for cloud computing platforms. Ideally, businesses should test the scalability of the application under a variety of network and application conditions to make it handles not only the current demands but also seamlessly scale to handle planned or unplanned spikes in demand.

II. LITERATURE REVIEW

A. *Understanding Performance, Scale and Throughput*

Because the terms performance, scale, and throughput are used in a variety of ways when deliberating computing, it is useful to scrutinize their typical meanings in the context of cloud computing infrastructures.

- Performance: Performance [6][10] is usually tied to an application’s capabilities within the cloud infrastructure itself. Limited disk space, bandwidth, CPU cycles, memory and network connections can cause poor performance. Often, a combination of lack of resources causes poor application performance. Sometimes poor performance is the outcome of an application design that does not properly allocate its processes across available cloud resources.
- Throughput: [6][10]The effective rate at which data is transferred from one point to the other on the cloud is throughput. In other words, throughput is a dimension of raw speed. While speed of moving or processing data can

positively improve system performance, the system is only as fast as its sluggish element. A system that deploys ten gigabit Ethernet yet its server storage can access data at only one gigabit successfully has a one gigabit system.

- Scalability: [6][10]The search for persistently improving system performance through software and hardware throughput gains is crushed when a system is flooded by multiple, coinciding demands. That 10 gigabit pipe slows considerably when it serves hundreds of requests rather than a dozen. The only way to renovate higher effective throughput (and performance) in such a “swamped resources” scenario is to scale—add more of the resource that is overloaded. For this reason, the ability of a system to easily scale when under stress in a cloud environment is vastly more useful than the overall throughput or aggregate performance of individual components.

B. Approaches to be discussed to make cloud computing more effective

1) Scalability - the best solution to increasing and maintaining application.
2) Green cloud computing Architecture.
3) Virtual machine migration.
4) Power management with Dynamic Voltage & Frequency Scaling (DVFS).

1) Scalability is the best solution [6] to increasing and maintaining application performance in cloud computing environment. According to a joyent inc. a complete cloud management solution scalability is the only best solution for increasing and maintaining application performance in cloud computing environment.

Horizontal and Vertical Scalability

When increasing assets on the cloud to restore or expand application performance, managers can scale either horizontally (out) or vertically (up), based on the nature of the resource constraint. Vertical scaling (up) requires adding more resources to the same computing pool—for example, adding more disk, RAM or virtual CPU to handle an increased application load. Horizontal scaling [11] (out) requires the addition of more machines or devices to the computing platform to handle the increased demand. This is represented in the transition from Figure 1 to Figure 2, below.

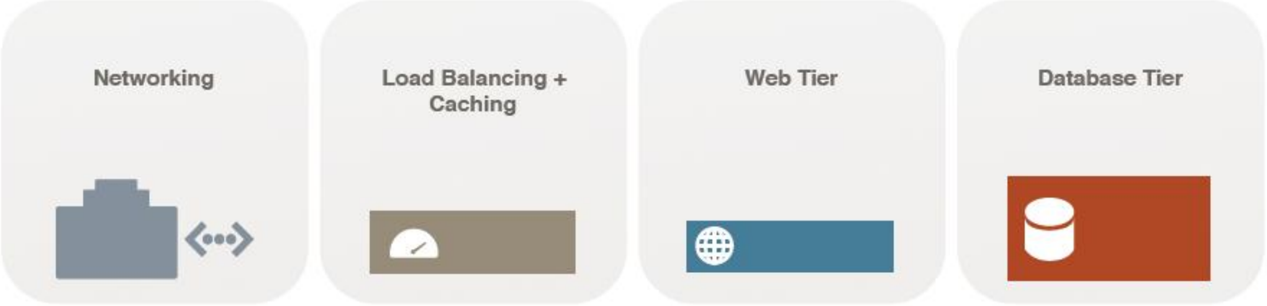

Figure 1: Basic, Single Silo, n-tier architecture

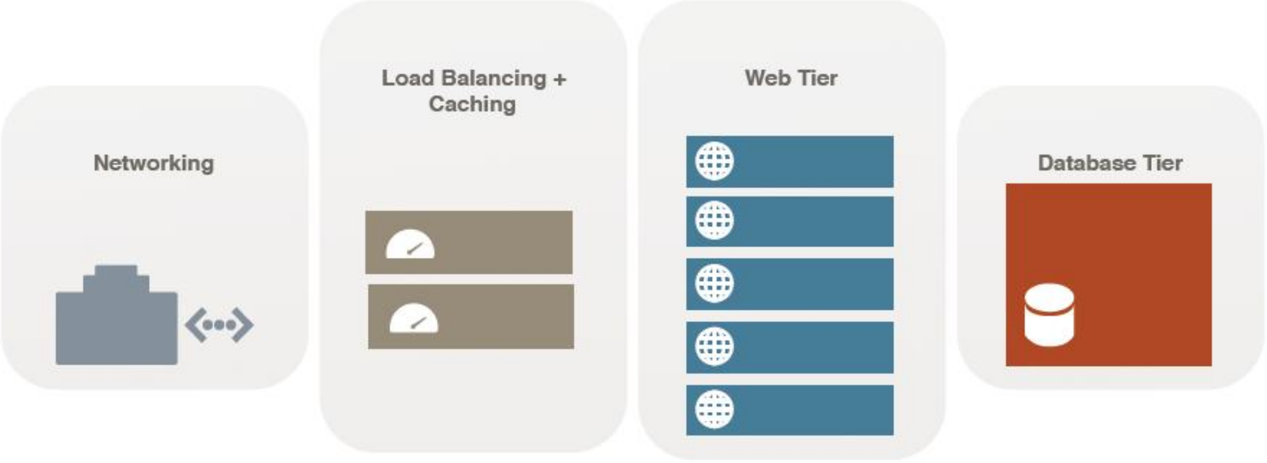

Figure 2: Horizontally scaled load balancing and web-tier. Vertically scaled database tier.

Figure 1, 2 [6]

Vertical scaling [11] can handle most impulsive, momentary peaks in application demand on cloud infrastructures since they are not typically CPU intensive tasks. Sustained increases in demand however requires horizontal scaling and load balancing to restore and preserve peak performance. Horizontal scaling is also manually concentrated and time consuming, requiring a technician to add machinery to the customer’s cloud configuration. Manually scaling to meet a sudden crowning in traffic may not be creative—traffic may settle to its pre-peak levels before new provisioning can come on line.

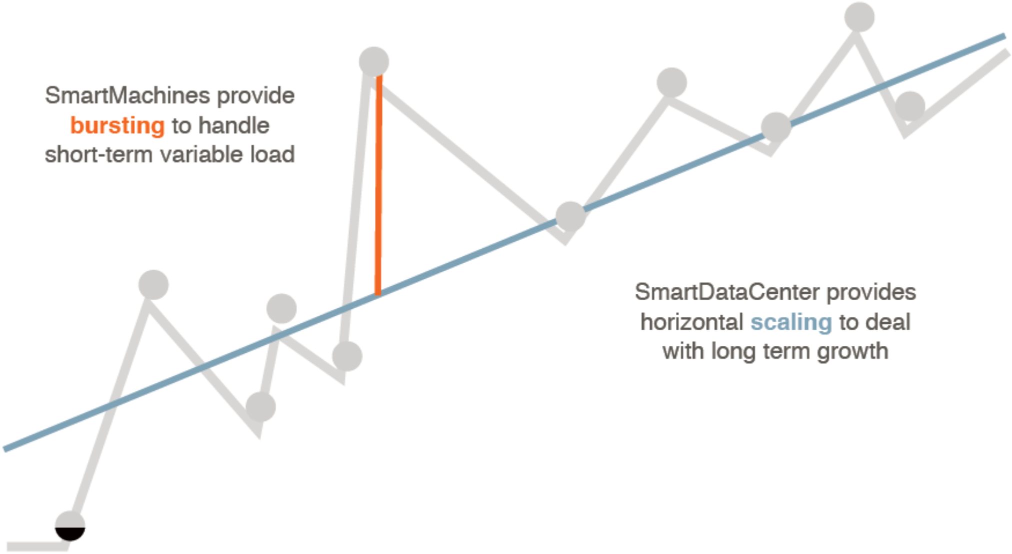
SmartMachines provide bursting to handle short-term variable load
SmartDataCenter provides horizontal scaling to deal with long term growth

Figure 3: Scaling – Vertical vs Horizontal

Businesses may also find themselves live through more gradual intensifications in traffic. Here, provisioning additional resources provides only momentary relief as resource demands continue to rise and outdo the newly provisioned resources.

Administrative and Geographical Scalability

In administrative and Geographical Scalability[6] accumulation of computing apparatuses or virtual resources is a logical way to scale and expand performance, few companies comprehend that the increase in resources may also require an increase in administration, predominantly when deploying horizontal scaling. In essence, a scaled increase in virtual or hard resources habitually necessitates a corresponding surge in administrative time and costs. This administrative rise may not be a one-time configuration request as more resources require repeated monitoring, maintenance and backups. Smart Machines provide bursting to manage short-term variable load. 'Smart Data Center' provides horizontal scaling to manage with long term growth.

Companies with business critical cloud applications might also deliberate geographical scaling as a means to more widely distribute application load stresses or as a way to move application access closer to spread communities of users or customers. Geographical scaling of resources in combination with real time replication of data pools is another means of adding fault tolerance and catastrophe recovery to cloud based data and applications. Geographical scaling may also be necessary in environments where it is unreasonable to host all data or applications in one central location.

Practical and Theoretical Limits of Scale

While scalability is the most operative approach for solving performance issues in cloud infrastructures, practical & theoretical boundaries prevent it from ever becoming an exponential solution. Practically speaking, most companies cannot pledge an infinite sum of money, people, or time to cultivate application performance. Cloud vendors also may have a limited amount of personnel, experience or bandwidth to address client application performance. Every computing infrastructure is bound by a certain level of complexity and scale, not the slightest of which is power, administration, and bandwidth, necessitating geographical dispersal.

Addressing Application Scalability

For a cloud computing platform to meritoriously host business data and applications it must accommodate a wide range of performance characteristics and network demands. Storage, memory, CPU and network bandwidth all come into play at countless times during typical application use. Application switching, for example, places demands on the CPU as one application is closed, flushed from the registers and another application is loaded. If these applications are large and complex, they put a more demand on the CPU. Portion files from the cloud to connected users stresses a number of resources, including disk drives, drive controllers, & network when transporting the data from the cloud to the user. File storage itself chomps resources not only in the form of physical disk space, but also disk directories and metafile systems that guzzle RAM and CPU cycles when users either access or upload files into the storage system. As these examples illustrate, applications can benefit from both horizontal and vertical scaling of resources on demand, yet truthfully dynamic scaling is not possible on most cloud computing infrastructures. Therefore, one of the most common and costly reactions to scaling issues by vendors is to over-provision client installations to house a wide range of performance issues.

Application Development to Improve Scalability

One practical means for addressing application scalability and to reduce performance bottlenecks [11] is to slice applications into distinct silos. Web-based applications are supposedly stateless, and therefore notionally easy to scale—all that is needed is more CPU, memory, storage, and bandwidth to manage them as was depicted in Figure 2. However, in practice Web-based applications are not stateless. They are accessed through a network connections that involves an IP addresses that is fixed and hence stateful, and they connect to data storage (either disk or database) which preserves logical state as well as requires hardware resources to execute. Balancing the interface between stateless and stateful elements of a Web application requires careful architectural thoughtfulness and the use of tiers and silos to allow some form of horizontal resource scaling. To leverage the most from resources, application developers can break applications into discrete tiers—state or stateless processes—that are implemented in various resource silos. Figure 4 depicts slicing an application into two silos identified by their DNS name. By isolating state and stateless operations and provisioning consequently, applications and systems can run more competently and with higher resource utilization than under a more common scenario.

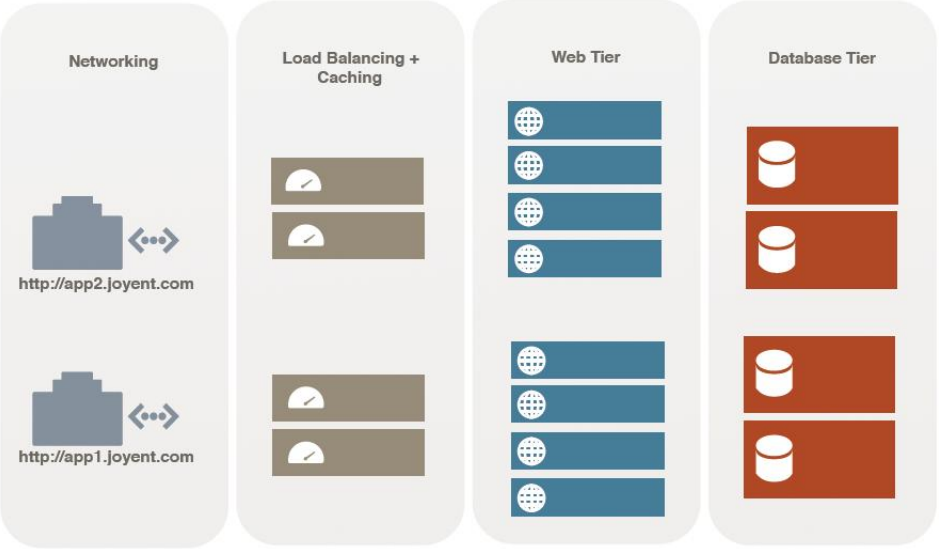

Figure4[6]:Slicing an application.

2) Architecture of a green cloud computing

Green Cloud computing is intended to achieve not only efficient processing and utilization of computing infrastructure, but also lessen energy consumption. This is vital for ensuring that the future growth of Cloud computing is justifiable. Otherwise, Cloud computing with increasingly universal front-end client devices interacting with back-end data centers will cause a massive intensification of energy usage. To tackle this problem, data center resources need to be operated in an energy efficient manner [12] to drive Green Cloud computing. In actual, Cloud resources need to be assigned not only to satisfy QoS requirements specified by users via Service Level Agreements (SLA), but also to reduce energy usage. Figure.5 shows the high-level architecture for supporting energy efficient service allocation in Green Cloud computing Architecture[7][9]. There are basically four main entities involved:

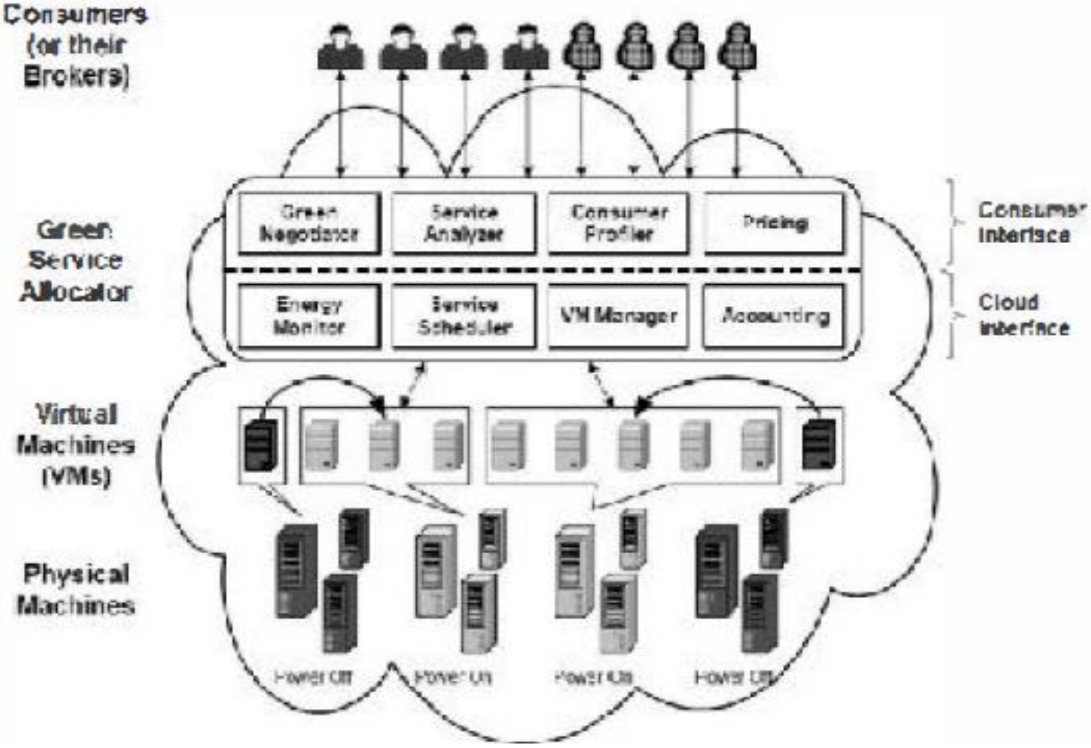

Figure.5: Architecture of a green cloud computing environment [7]

© 2014, IJARCSSE All Rights Reserved

a) *Consumers/Brokers:* Cloud customers or their brokers submit service requests from anywhere in the world to the Cloud. It is important to notice that there can be a difference between Cloud consumers and users of deployed services. For instance, a consumer can be a company set up a Web application, which presents varying workload according to the number of users accessing it.

b) *Green Resource Allocator*: [7] Acts as the interface between the Cloud infrastructure and clients. It necessitates the interaction of the following components to support energy efficient resource management:

1. *Green Negotiator:* Conveys with the consumers/brokers to finalize the SLA with specified prices and penalties (for violations of SLA) between the Cloud provider and consumer depending on the consumer's QoS necessities and energy saving schemes.
2. *Service Analyzer:* Interprets and analyses the service requirements of a submitted request before deciding whether to accept or discard it. Hence, it needs the modern load and energy information from VM Manager and Energy Monitor respectively.
3. *Consumer Profiler:* Gathers specific characteristics of consumers so that important consumers can be granted special rights and prioritized over other consumers.
4. *Pricing:* Decides how service requests are charged to manage the supply and demand of computing resources and facilitate in prioritizing service allocations effectively.
5. *Energy Monitor:* Observes and determines which physical machines to power on/off.
6. *Service Scheduler:* Assigns requests to VMs and determines resource privileges for assigned VMs. It also decides when VMs are to be spin-up or down to meet demand.
7. *VM Manager:* Keeps track of the availability of VMs and their resource rights. It is also in charge of moving VMs across physical machines.
8. *Accounting:* Maintains the actual usage of resources by requests to compute usage charges. Historical usage data can also be used to improve service allocation decisions.

c) *VMs:* Multiple VMs can be dynamically started and stopped on a single machine to meet accepted requests, hence providing unwavering elasticity to organize various partitions of resources on the same physical machine to different specific requirements of service requests. Multiple VMs can also in parallel run applications based on different operating system environments on a single physical machine. In addition, by dynamically moving VMs across physical machines, workloads can be fused and unused resources can be put on a low-power state, turned off or designed to operate at low-performance levels (e.g., using DVFS) in order to save energy.

d) *Physical Machines:* The core physical computing servers provide hardware infrastructure for forming virtualized resources to meet service demands.

3) Virtual Machine Migration

Among all these approaches, Virtual Machine (VM) expertise begins to emerge as a focus of research and deployment. Virtual Machine (VM) technology (such as VMWare, Microsoft Virtual Servers, Xen, and the new Microsoft Hyper-V technology etc), enables several OS environments to co-exist on the same computer, in strong seclusion from each other. VMs share the predictable hardware in a secure mode with exceptional resource management capacity, while each VM is hosting its own operating system and programs.

Hence, VM platform can enable server-consolidation and co-located facilities [13]. Virtual machine migration,[9] which is used to displace a VM across physical computers, has operated as a main approach to attain improved energy efficiency. In doing so, server consolidation via VM migrations permits more computers to be turned off. Generally, there are two variations [14]: steady migration and live migration. The former moves a VM from one host computer to another by resting the initially used server, copying its memory contents, and then resuming it on the target. The second performs the same logical functionality but without the need to pause the server for the transition. In general when executing live migrations the domain continues its typical activities and from the user's perspective—the migration should be invisible. It shows great prospective of using VM and VM migration technology to competently manage workload amalgamation, and therefore improve the total IDC power efficiency [9].

VM Power Management & Migration

In IDC, there are two types of Virtualization technologies that are studied a lot lately. One is full-virtualization technology, such as VMWare [15]. Full-virtualization, otherwise known as inherent virtualization, uses a virtual machine that adjudicates between the guest operating systems and the native hardware. VMM facilitates between the guest operating systems and the bare hardware. Certain sheltered instructions must be trapped and handled within the hypervisor because the primary hardware isn't owned by an operating system but is instead shared by it through the hypervisor. On the other hand, para-virtualization is a very popular technique that has some resemblances to full virtualization. This method uses a hypervisor for shared admission to the underlying hardware but assimilates virtualization-aware code into the operating system itself. This approach removes the need for any recompilation or trapping as the operating systems themselves collaborate in the virtualization process. A typical para virtualization product is Xen. While various administration strategies have been developed to effectually reduce server power consumption by transitioning hardware components to lower-power states, they cannot be straight applied to today's data centers that rely on virtualization technologies.

Nathuji et al. [14] have proposed an online power administration to support the isolated and sovereign operation assumed by VMs running on a virtualized platform and universally coordinate the diverse power management strategies applied by the VMs to the virtualized resource. They utilize the "Virtual Power" to characterize the 'soft' versions of the hardware power state, to enable the deployment of the power management policies. In order to mapthe 'soft' power state to the actual changes of the fundamental virtualized resource, the Virtual Power Management (VPM) state, mechanisms, channels and rules are implemented as the multiple system level notion. In the early research, the Collective project [9], has designed VM migration as a tool to deliver mobility to users who work on different physical machines at different times. This solution aims at the course of transferring an OS instance through slow links and longtime spans. With a set of enrichment work to reduce the image size, it will stop the running of the VM through the migration duration.

Live Migration

For performance-sensitive applications, VM live migration offers great benefits we attempt to augment the utilization of available resources (e.g., CPU). In VM live migration, a VM is migrated from on physical server to another while uninterruptedly running, without any visible effects from the point of view of the end users. During this procedure, the memory of the virtual machine is iteratively copied to the destination without stopping its execution.

The halt of around 60–300 ms is mandatory to perform the final synchronization before the virtual machine begins executing at its final destination, providing an impression of seamless migration. However, with the old-fashioned VM migration technology, which stops the running VM during the migration, will cause the failure to meet the Service Level Agreement (SLA) guarantees, especially in the response time delicate computing.

4) Power and Energy Management

To comprehend power and energy management instruments it is essential to clearly distinguish the background terms. Electric current is the flow of electric charge dignified in Amperes (Amps). Amperes define the amount of electric charge transferred by a circuit per second. Power and energy can be defined in terms of work that a system performs. Power is the rate at which the system performs the work, while energy is the total amount of work performed over a period of time. Power and energy are measured in watts (W) and watt-hour (Wh) respectively. Work is done at the rate of one watt when one Ampere is transferred through a potential difference of one volt. A kilowatt-hour (kWh) is the amount of energy equivalent to a power of 1 kilowatt (1000 watts) running for 1 hour. Formally, power and energy can be defined as in (1) and (2).

$$P=W/T \quad \text{.... (1)}$$

$$E= P.\,T \quad \text{.... (2)}$$

Where P is power, T is a period of time, W is the total work performed in that period of time, and
E is energy. The variance between power and energy is very important, because reduction of the power consumption does not constantly reduce the consumed energy.

Static and Dynamic Power Consumption

The main power consumption in Complementary Metal-Oxide-Semiconductor (CMOS) circuits comprises static and dynamic power[8]. The static power consumption, or leakage power, is instigated by leakage currents that are extant in any active circuit, autonomous of clock rates and usage scenarios. This static power is chiefly determined by the type of transistors and process technology. Decrease of the static power necessitates improvement of the low-level system design; Dynamic power consumption is created by circuit activity (i.e. transistor switches, changes of values in registers, etc.) and governed by mainly on a specific usage scenario, clock rates, and I/O activity. The causes of the dynamic

Taxonomy of Power / Energy Management in Computing Systems

Large volume of study work has been done in the area of power and energy-efficient resource management in computing structures. As power and energy management techniques are diligently connected, from this point we will refer to them as power management. As shown in Figure 8, from the high level power management practices can be divided into static and dynamic. From hardware point of view, Static Power Management (SPM) contains all the optimization methods that are applied at the design time at the circuit, architectural, logic and system levels [17].

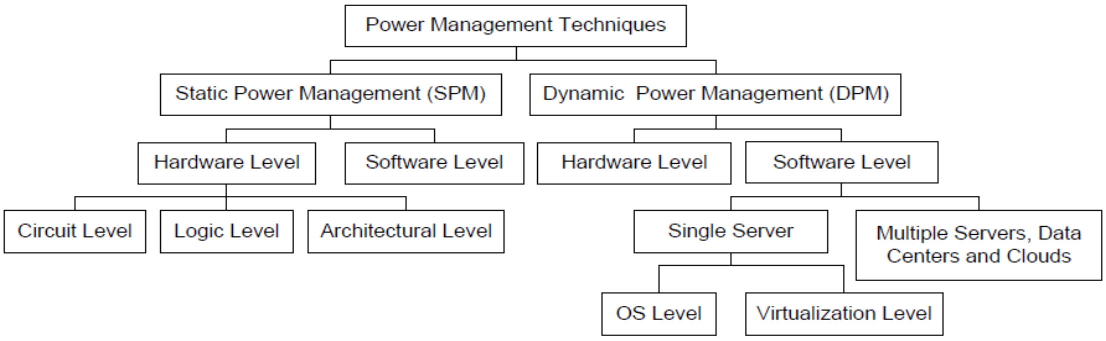

Figure 8. High level taxonomy of power and energy management.[8]

Circuit level optimizations are focused on the saving of switching activity power of individual logic gates and transistor level combinational circuits by the application of a complex gate design and transistor sizing. Optimizations at the logic level are aimed at the swapping activity power of logic level combinational and sequential circuits.

Hardware and Firmware Level

As shown in Figure 9, DPM techniques applied at the hardware and firmware level can be largely divided into two categories: Dynamic Component Deactivation (DCD) and Dynamic Performance Scaling (DPS). DCD techniques are erected upon the idea of the clock gating of parts of an electronic component or complete restricting during periods of inactivity.

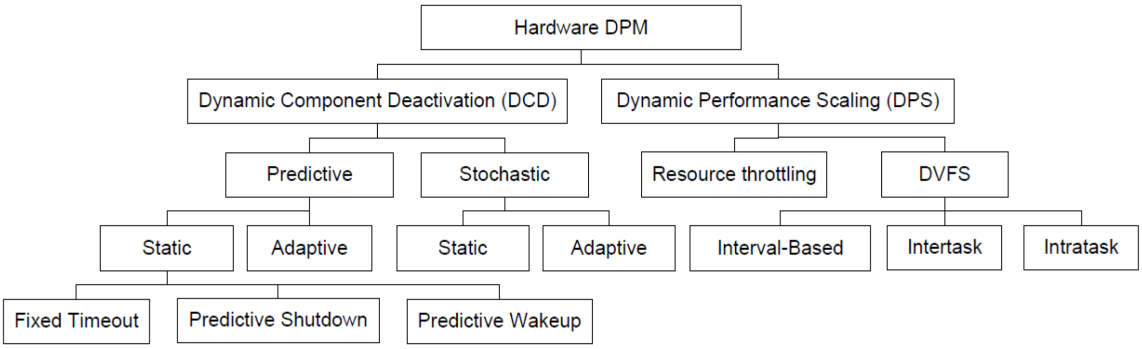

Figure 9. DPM techniques applied at the hardware and firmware levels.[8]

The problem could be easily resolved if transitions between power states would cause insignificant power and performance overhead. However, transitions to low-power states usually lead to added power consumption and interruptions caused by the re-initialization of the components. For example, if entering a low-power state requires shut-down of the power supply, returning to the active state will cause a delay involving of: turning on and stabilizing the power supply and clock; re-initialization of the system; and restoring the environment [16]. In the case of non-negligible transitions, actual power management turns into a tough on-line optimization problem. A shift to low-power state is worthwhile only if the period of inactivity is longer than the accumulated delay of transitions from and into the active state, and saved power is higher than required to reinitialize the component.

Dynamic Component Deactivation (DCD)

Computer components that do not provision performance scaling and can only be disabled require techniques that will leverage the workload variability and disable the component when it is idle. The problem is minor in the case of a negligible transition overhead. However, in reality such transitions lead not only to delays, which can degrade performance of the system, but to additional power draw. Therefore, to achieve efficiency a shift has to be done only if the idle period is adequately long to cover the transition overhead. In most real-world systems there is a limited or no knowledge about the future workload. Therefore, a prediction of an effective transition has to be done bestowing to historical data or some system model. A large volume of study has been done to develop efficient methods to solve this problem. As shown in Figure 9, the proposed DCD techniques can be divided into predictive and stochastic.

Dynamic Performance Scaling (DPS)

Dynamic Performance Scaling (DPS)[8] includes different methods that can be applied to computer components supporting dynamic adjustment of their performance proportionally to the power consumption. Inspite of complete deactivations, some components, such as CPU, allow slow decrease or increases of the clock frequency along with the adjustment of the supply voltage in cases when the resource is not utilizing of its full capacity. This idea lies in the roots of the broadly adopted Dynamic Voltage and Frequency Scaling (DVFS) technique.

1) Dynamic Voltage and Frequency Scaling (DVFS)

Although the CPU frequency can be attuned separately, frequency scaling by itself is rarely worthwhile as a way to preserve switching power. Saving the most power requires dynamic voltage scaling too, because of the V2 factor and the fact that modern CPUs are strongly enhanced for low voltage states. Dynamic voltage scaling is typically used in combination with frequency scaling, as the frequency that a chip may run at is linked to the operating voltage. The efficiency of some electrical components, such as voltage regulators, reduces with a temperature rise, so the power used may increase with temperature. Since increasing power use may raise the temperature, rise in voltage or frequency may increase the system power demand, and vice-versa. DVFS reduces the number of instructions a processor can issue in a specified amount of time, thus decreasing the performance. This, in turn, increases run time for program section which are suitably CPU-bound. Hence, it creates trials of providing optimal energy / performance control, which have been broadly investigated by scientists in recent years. Some of the study works will be reviewed in the following sections.

Although the application of DVFS may seem to be upfront, real-world systems increase many complexities that have to be considered. First of all, due to composite (complex) architectures of modern CPUs (i.e. pipelining, multi-level cache, etc.), the estimate of the required CPU clock frequency that will meet application’s performance requirements is not trivial. Another problem is that on the other hand of the theory, power consumption by a CPU may not be quadratic to its supply voltage. For example some architectures may contain several supply voltages that power dissimilar parts of the chip, and even if one of them can be reduced, overall power consumption will be dominated by the larger supply voltage. Moreover, execution time of the program running on the CPU may not be inversely proportional to the clock frequency, and DVFS may result in non-linarites in the execution time [16]. For example, if the program is memory or I/O restricted, CPU speed will not have a effect on the execution time. Moreover, slowing down the CPU may lead to changes in the sequences in which tasks are scheduled. In summary, DVFS can provide considerable energy savings; however, it has to be applied sensibly, as the result may significantly vary for different hardware and software system architectures.

Approaches that apply DVFS to decrease energy consumption by a system can be divided into interval-based, intertask and intratask [8]. Interval-based algorithms are like to adaptive predictive DCD approache in that they also utilize information of the previous periods of the CPU activity . Depending on the use of the CPU during preceding intervals, they predict the consumption in the near future and properly adjust the voltage and clock frequency. Wierman et al. [19] and Andrew et al. [20] have showed investigative studies of speed scaling algorithms in processor sharing systems. They have proved that no online energy-proportional speed scaling algorithm can be superior than 2-competitive comparing to the offline optimal algorithm. Moreover, they have originate that difficulty in the design of speed scaling algorithms does not provide significant performance advances; however, it dramatically improves toughness to errors in estimation of workload parameters. Intertask method instead of relying on rough grained information on the CPU utilization, discriminate different tasks running in the system and assign them different speeds. The problem is easy to solve if the workload is known a priori or constant over all the period of a task execution. However, the problem becomes non-trivial when the workload is unequal. In contrast to intertask, intratask methods influence fine grained information about the construction of programs and adjust the processor frequency and voltage within the tasks. Such guidelines can be applied by splitting a program execution into timeslots and allocating different CPU speeds to each of them. Another way is to implement them at the compiler level. This kind of approaches consumes compiler’s knowledge of a program’s structure to make implications about possible periods for the clock frequency reduction.

III. CONCLUSIONS

Cloud computing is emerging as a significant shift as today's organizations which are facing extreme data overload and sky rocketing energy costs. In this Report, we propose various approaches Scalability, Green Cloud architecture, Virtual Migration and Dynamic Voltage and Frequency Scaling (DVFS) which can help consolidate workload and achieve significant energy saving for cloud computing environment, at the same time, assurances the real-time performance for many performance-sensitive applications. The Green Cloud influences the state-of-the-art live virtual machine migration technology to accomplish these goals. In the future, there are still a number of research activities that we plan to carry out, which could enhance the performance of Green Cloud and take solid value to users to get their business goals and their social obligation in Green IT. Further studies should be given to explore whether utility-based Methodology can be used to converse performance-power tradeoffs between the OS and the application/middleware.

ACKNOWLEDGMENT

I Amanpreet Kaur wish to acknowledge Deepa Gupta and Deepak Kumar Verma for their endless support and guidance on putting this research together.

REFERENCES

[1] NIST (Authors: P. Mell and T. Grance), "The NIST Definition of Cloud Computing (ver. 15)," National Institute of Standards and Technology, Information Technology Laboratory (October 7 2009).

[2] NIST(Authors: Dr. Thomas Cynkin) Joint Cloud and Big Data Workshop January 15-17, 2013.

[3] Cloud Computing", 391h IEEE international Conference on Parallel Processing Workshops 2010, IEEE Computer Society, pp. 275-279.

[4] http://en.wikipedia.org/wiki/Cloud_computing.

[5] Yashpalsingh and Jadeja Kirit Modi “Cloud Computing - Concepts, Architecture and Challenges” 2012 International Conference on Computing, Electronics and Electrical Technologies [ICCEET].

[6] A Joyent White Paper” Performance and Scale in Cloud Computing” By joyent inc.- san Francisco.

[7] R.Yamini ,Assistant Professor” Power Management in Cloud Computing Using Green Algorithm” IEEE-International Conference On Advances In Engineering, Science And Management (ICAESM -2012) March 30, 31, 2012.

[8] Anton Beloglazov1, Rajkumar Buyya1, Young Choon Lee2, and Albert Zomaya2” A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems”.

[9] Liang Liu, Hao Wang, Xue Liu, Xing Jin, WenBo He, QingBo Wang, Ying Chen “GreenCloud: A New Architecture for Green Data Center” ICAC-INDST’09, June 16, 2009, Barcelona, Spain.

[10] Mouline, Imad. “Why Assumptions About Cloud Performance Can Be Dangerous.” Cloud Computing Journal. May, 2009 www.cloudcomputing.sys-con.com/node/957492

[11] Nolle, Tom. “Meeting performance standards and SLAs in the cloud.”

[12] A. Berl, E. Gelenbe, M. di Girolamo, G. Giuliani, H.de Meer, M.-Q. Dang, and K. Pentikousis. EnergyEfficient Cloud Computing. The Computer Journal,53(7), September20 1O.

[13] A. Whitaker, M. Shaw, S. D. Gribble, "Lightweight Virtual Machines for Distributed and Networked Applications".Technical Report 02-02-01, University of Washington, 2002.

[14] R.Nathuji, K. Schwan, "VirtualPower: coordinated power management in virtualized enterprise systems", ACM Symposium on Operating Systems Principles, Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, 2007.

[15] VMWare, VMWare Inc. http://www.vmware.com

[16] L. Minas and B. Ellison, Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers. Intel Press, Aug. 2009.

[17] S. Devadas and S. Malik, ―A survey of optimization techniques targeting low power VLSI circuits,‖ in Proceedings of the 32nd ACM/IEEE Conference on Design Automation, 1995, pp. 242–247.

[18] V. Tiwari, P. Ashar, and S. Malik, ―Technology mapping for low power,‖ in Proceedings of the 30th Conference on Design Automation, 1993, pp. 74–79.

[19] A. Wierman, L. L. Andrew, and A. Tang, ―Power-aware speed scaling in processor sharing systems,‖ in Proceedings of the 28th Conference on Computer Communications (INFOCOM 2009), Rio, Brazil, 2009.

[20] L. L. Andrew, M. Lin, and A. Wierman, ―Optimality, fairness, and robustness in speed scaling designs,‖ in Proceedings of ACM International Conference on Measurement and Modeling of International Computer Systems (SIGMETRICS 2010), New York, USA, 2010.

© 2014, IJARCSSE All Rights Reserved